VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

Zayd Ahmed (1BM21CS254)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Oct 2022-Feb 2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED JAVA PROGRAMMING**" carried out by **Zayd Ahmed(1BM21CS254),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Java Lab **- (22CS3PCOOJ)** work prescribed for the said degree.

Name of the Lab-Incharge
Designation
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index Sheet

Course Outcome

CO1	Apply the knowledge of java concepts to find the solution for a given solution
CO2	Analyze the given java application for correctness
CO3	Develop Java programs for a given requirement
CO4	Conduct practical experiments for demonstrating features of java

LAB PROGRAM 1:

Q: Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

4) Java code to solve Quadratic equation

```
import java.util.Scanner;
class Equation
{
  public static void main (String args[])
  {
    System.out.println("enter the coefficients");
    Scanner sc = new Scanner(System.in);
    double a = sc.nextint();
    if (a==0)
    {
      System.out.println("a cannot be zero");
    }
  else
  {
    double b = sc.nextint();
  double c = sc.next int();
  double z = b*b -4*a*c;
  EquationCheck ob = new EquationCheck();
  if (z<0)
  {
    System.out.println("there are no real solutions");
  }
  else if (z==0)
  {
    System.out.println("the solutions are real and equal")
    ob.check(a,b,c);
    ob.display();
  }
```

```
else
{
 System.out.println("Solutions are real and distinct");
 ob.check(a,b,c);
 ob.display();
}
}
}
}
class EquationCheck
{
 double a, b, c, x1, x2;
 void check(double a, double b, double c)
 {
  this.a = a;
  this.b = b;
  this.c = c;
  double z = Math.pow(b*b - 4*a*c, 0.5);
  x1 = (-b - z)/(2*a);
  x2 = (-b + z)/(2*a);
 }
 void display()
 {
  System.out.println(x1);
  System.out.println(x2);
 }
}
```

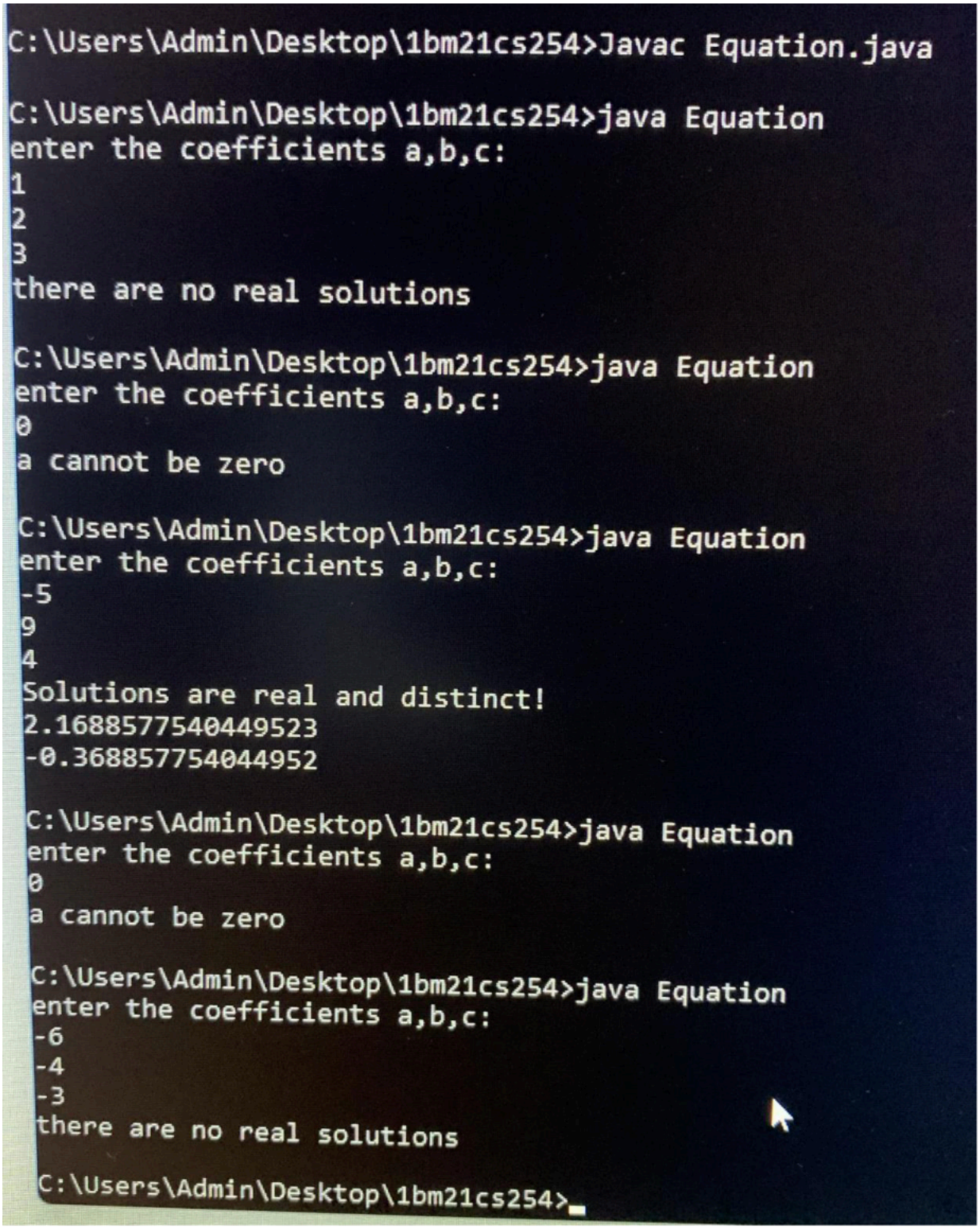
Output

17/11/2022

```
enter the coefficients 1 -5 6
The solutions are real and distinct    2.0    3.0
```

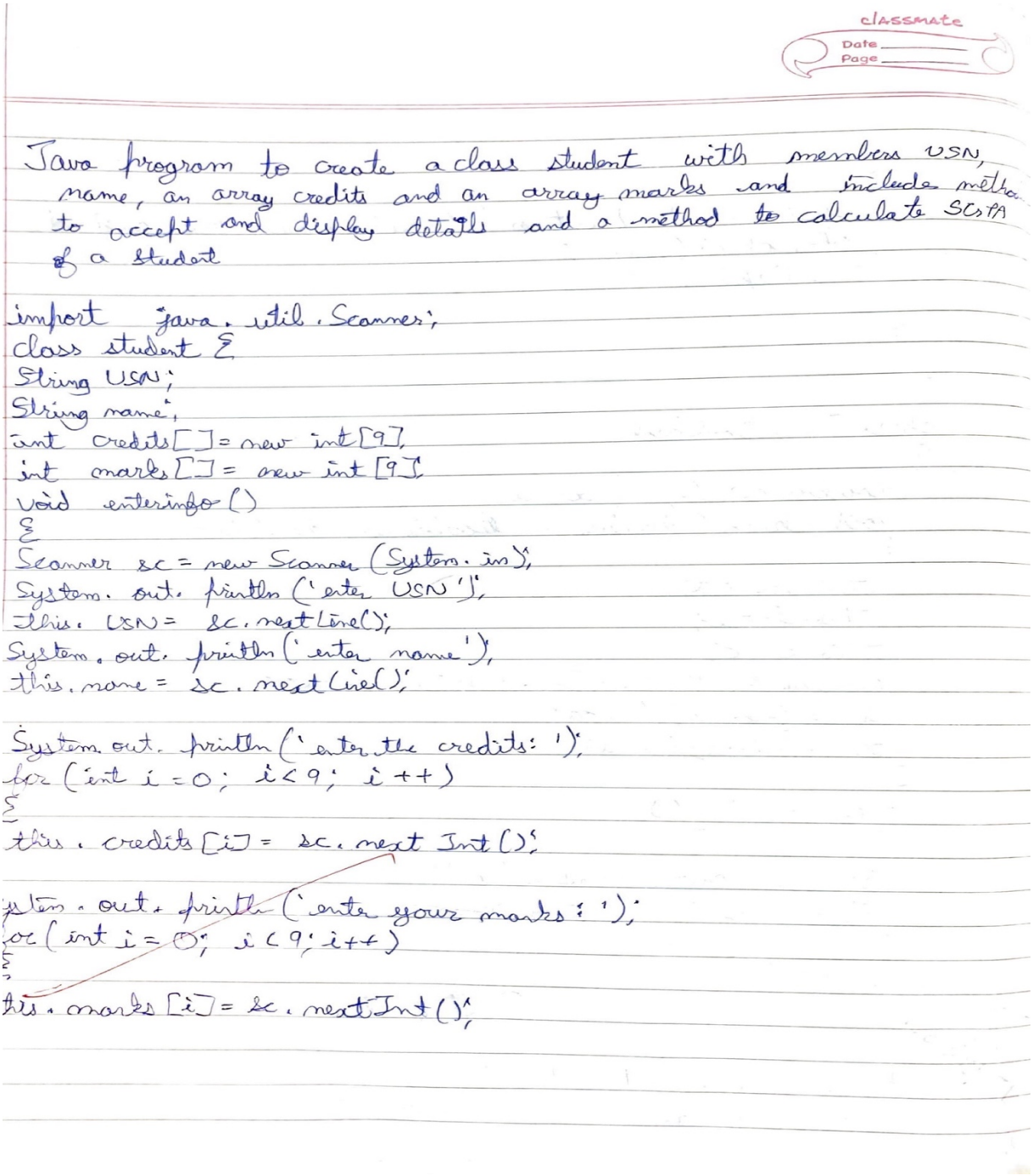
—x—

Output:


C:\Users\Admin\Desktop\1bm21cs254>Javac Equation.java
C:\Users\Admin\Desktop\1bm21cs254>java Equation
enter the coefficients a,b,c:
1
2
3
there are no real solutions
C:\Users\Admin\Desktop\1bm21cs254>java Equation
enter the coefficients a,b,c:
0
a cannot be zero
C:\Users\Admin\Desktop\1bm21cs254>java Equation
enter the coefficients a,b,c:
-5
9
4
Solutions are real and distinct!
2.1688577540449523
-0.368857754044952
C:\Users\Admin\Desktop\1bm21cs254>java Equation
enter the coefficients a,b,c:
0
a cannot be zero
C:\Users\Admin\Desktop\1bm21cs254>java Equation
enter the coefficients a,b,c:
-6
-4
-3
there are no real solutions
C:\Users\Admin\Desktop\1bm21cs254>

LAB PROGRAM 2:

Q: Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.



```
Java program to create a class student with members USN,
name, an array credits and an array marks and include metho
to accept and display details and a method to calculate SGPA
of a student

import java.util.Scanner;
class student {
String USN;
String name;
int credits[] = new int[9];
int marks[] = new int[9];
void enterinfo()
{
Scanner sc = new Scanner(System.in);
System.out.println("enter USN");
this.USN = sc.nextLine();
System.out.println("enter name");
this.name = sc.nextLine();

System.out.println("enter the credits:");
for(int i=0; i<9; i++)
{
this.credits[i] = sc.nextInt();

stem.out.println("enter your marks:");
or(int i=0; i<9; i++)
{
his.marks[i] = sc.nextInt();
```

```
void displayInfo()
{
System.out.println("\n is the student Info :-\n");
System.out.println("USN : "+ this.usn);
System.out.println("NAME : "+this.name);
System.out.println("Credits : ");
for(int i = 0; i < 9; i++)
{
System.out.println(this.marks[i]+ " ");
}
}
float calculateSGPA()
{
float sgpa;
float totalcreds = 0;
for(int i = 0; i <9; i++)
{
totalcreds += this.credits[i];
}
float gp = 0;
for(int i=0; i<9; i++)
{
gp += this.credits[i] * (((this.marks[i])/10)+1);
}
sgpa = gp/totalcreds;
return sgpa;
}
}
public class calc
{
public static void main(String args[])
```

```
{
Student s1 = new student();
S1. enterInfo();
s1. displayInfo();
float sgpa = s1. calculateSGPA();
System.out.println("\n SGPA: "+sgpa);
}
}
```

O/P

```
enter USN
1BM21CS254
enter name
Zayd Ahmed
enter credits
3 4 1 3 1 3 1
enter your marks
50 62 86 68 62 75 56 78 79
Below is the Student info
USN: 1BM21CS254
name: Zayd Ahmed
SGPA: 7.25
```

SSB
11/12/2024

Output:

```
C:\Users\Admin\Desktop\1bm21cs254>java calc
enter the USN
1BM21CS254
Enter the Name:
L.Lawliet
Enter the credits:
3 4 1 3 1 3 1 3 1
enter your marks:
50 62 86 68 62 75 56 78 79

Below is the Student Information:-

USN: 1BM21CS254
NAME: L.Lawliet
CREDITS:
50
62
86
68
62
75
56
78
79

SGPA: 7.25
```

LAB PROGRAM 3:

Q: Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

6) Create a Class which contains four members, name author price, num_pages. Include a constructor to set and get the details of the objects. Include a toString() method that could display the complete details of the book.

```
import java.util.Scanner;
class book {
  String author;
  String name;
  int pages;
  int price;
  book(String name1, String author1, int pages, int price1) {
      name = name1;
      author = author1;
      price = price1;
      num_pages = pages;
  }
  book() {
  void setdetails() {
  Scanner sc = new Scanner(System.in);
  System.out.print('enter name of book : ');
  name = sc.nextLine();
  System.out.print('enter authors name : ');
  author = sc.nextLine();
  System.out.print('enter price of book : ');
  price = sc.nextInt();
  System.out.print('enter num of pages : ');
  num_pages = sc.nextInt();
  System.out.println('Detail Set successfully :) ");
  }
```

```
void getdetails () {
    System.out.println ("NAME: " + name);
    System.out.println ("Author: " + author);
    System.out.println ("PRICE: " + price);
    System.out.println ("Num_Pages: " + num_pages);
}

public String toString () {
    return ("NAME: " + name + " AUTHOR: " + author + " PRICE = " + price +
    NUM_PAGE: " + num_pages);
}
}

class book1 {
    public static void main (String args[]) {
        int n;
        System.out.println ("enter the number of books: ");
        Scanner sc = new Scanner (System.in);
        n = sc.nextInt ();
        book b1[] = new book[n];
        for (int j = 0; j < n; j++) {
            System.out.println ("enter details");
            System.out.println ("1. set using constructor \n 2. set using meth
            int ch = sc.nextInt ();
            if (ch == 2) {
                b1[j] = new book ();
                b1[j].setdetails ();
            }
            else {
                b1[j] = new book ("Tinkle", "Anonymous", 20, 100);
            }
        }
```

```
for (int j = 0, j < n; j++) {
    System.out.println("\n Printing Book details");
    System.out.println("1. display using method \n 2: display using testing");
    int ch = sc.nextInt();
    if (ch == 1) {
        b1[j].getdetails();
        System.out.println();
    }
    else {
        String details = b1[j].toString();
        System.out.println(details);
    }
  }
 }
}
```

O/P

```
enter the no. of books: 2
enter details
1. set using constructor
2. set using method
2
enter name of book: lmao
enter authors name: fdefde
enter price of book: 11
enter num of pages: 1
Details set successfully :)
enter details
1. set using constructor
2. set using method
1
```

Printing book details
1. display using method
2. display using to String
1
NAME: lmao
AUTHOR: fdefde
PRICE: 220
NUM_PAGE: 20

Printing book details
1. display using method
2. display using to String
2
NAME: TINKLE AUTHOR: ANONYMOUS PRICE: 20 PAGES: 100

550
11/12/202

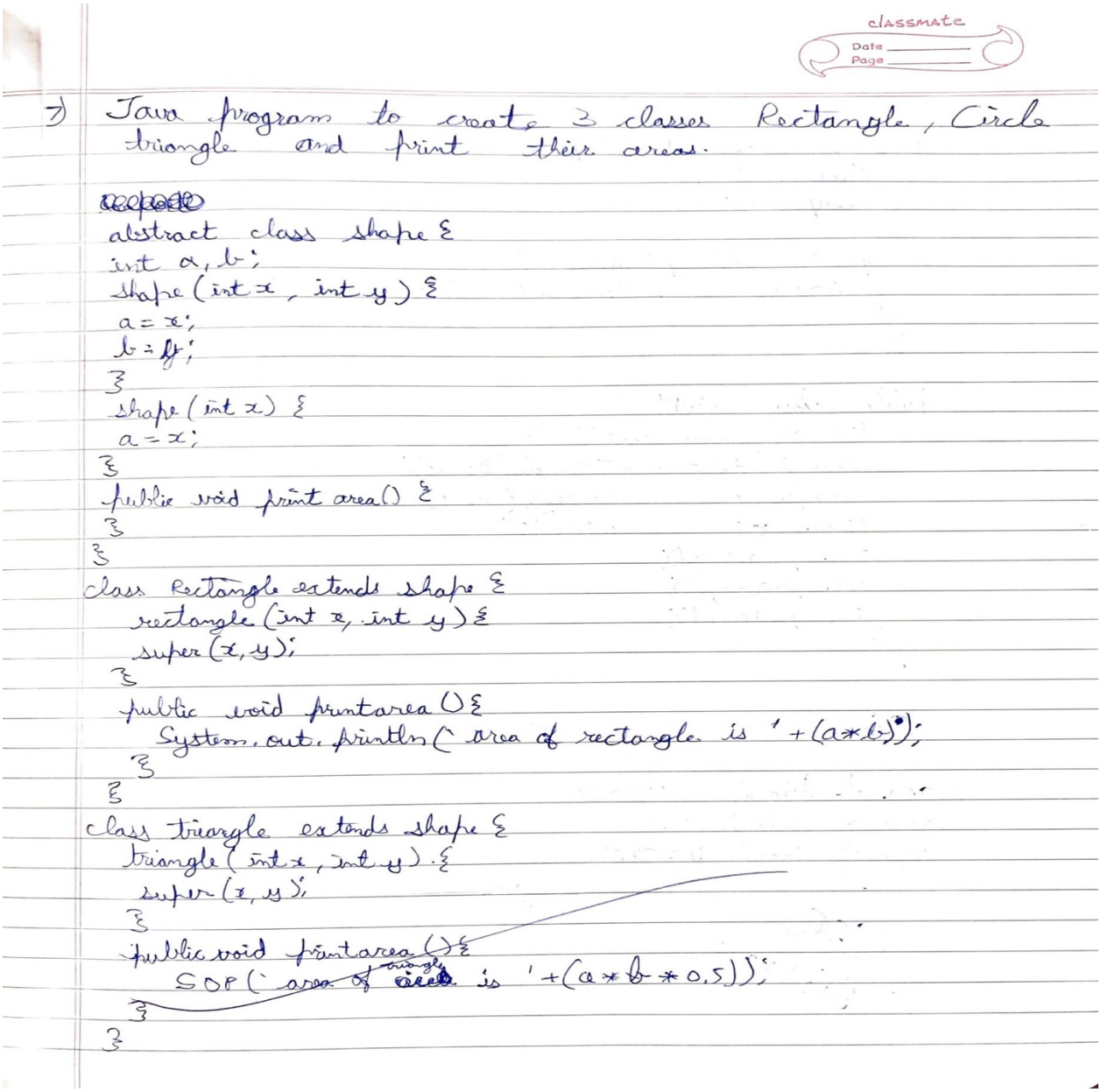
—x—

Output:

```
C:\Users\Admin\Desktop\1722>java book1
enter the number of books:
2
ENTER DETAILS
1:set using constructor
2:set using method
2
enter name of book: lmao
enter author's name: rfgg
enter price of book: 22
enter num of pages: 22
DETAILS SET SUCCESSFULLY :)
ENTER DETAILS
1:set using constructor
2:set using method
1

PRINTING BOOK DETAILS
1:display using method
2:display using toString
1
NAME: lmao
AUTHOR: rfgg
PRICE: 22
NUM_PAGES: 22


PRINTING BOOK DETAILS
1:display using method
2:display using toString
2
NAME: TINKLE AUTHOR: ANONYMOUS PRICE: 20 NUM_PAGES: 100
```

LAB PROGRAM 4:

Q: Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.



7) Java program to create 3 classes Rectangle, Circle triangle and print their areas.

```
abstract class shape {
int a, b;
shape (int x, int y) {
a = x;
b = y;
}
shape (int x) {
a = x;
}
public void print area() {
}
}
class Rectangle extends shape {
  rectangle (int x, int y) {
  super (x, y);
  }
  public void printarea () {
    System.out.println (" area of rectangle is " + (a*b));
  }
}
class Triangle extends shape {
  Triangle (int x, int y) {
  super (x, y);
  }
  public void printarea () {
    SOP (" area of triangle is " + (a * b * 0.5));
  }
}
```

```
class circle extends shape {
    circle (int x) {
     super(x);
    }
    public void printarea() {
      SOP(" area of circle is : "+(a*a*3.14));
    }
}

public class abstt {
    PSVM(String args[]) {
    Triangle t1 = new Triangle(30,40);
    rectangle r1 = new rectangle(10,20);
    circle c1 = new circle(5);
    t1.printarea();
    r1.printarea();
    c1.printarea();
    }
}
```

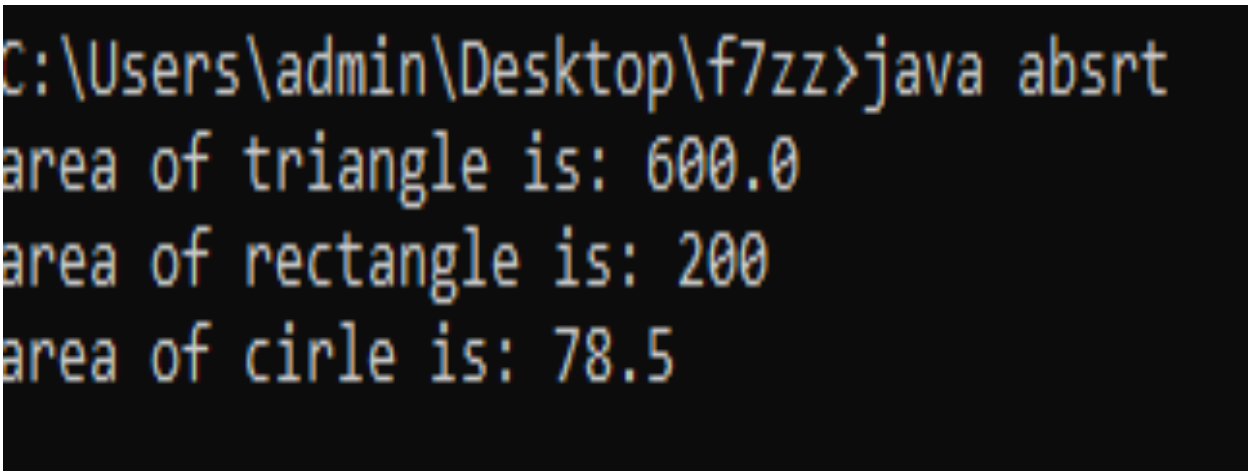
O/P

area of triangle is : 600.0
area of rectangle is : 200
area of circle is : 78.5

—X—

CSO
8/12/20

Output:


C:\Users\admin\Desktop\f7zz>java absrt
area of triangle is: 600.0
area of rectangle is: 200
area of cirle is: 78.5

LAB PROGRAM 5:

Q: Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements.

Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

8) Bank program for savings & current

```
import java.util.Scanner;

class account {
  String name;
  int account_num;
  String acc_type;
}
class sav_acct extends account {
double balance;
sav_acct(String n, int ac, String actype, Double bl) {
 name = n;
account_num = ac;
actype = acc_type;
 balance = bl;
}
Scanner sc = new Scanner(System.in);
void deposit(int val) {
    balance += val;
}
void display_bal() {
    System.out.println("balance is: " + balance);
}
void deposit_interest() {
 double int_rate = 0.05;
double time = 0;
SOP('enter the time period');
time = sc.nextDouble();
double amount;
```

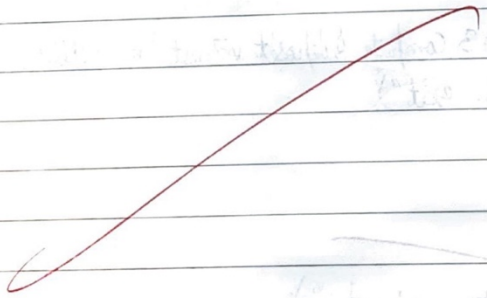
```
amount = balance * Math.pow((1+int_rate), time);
balance = amount;
}

void withdraw(int val) {
if (val > balance) {
  SOP("out of funds, withdraw lesser");
}
else {
  balance -= val;
  SOP("withdrawal successful");
  SOP("new balance:" + balance);
}
}

void check_min() {
Double min_bal = 1000.00;
Double penalty = 100.00;
if( balance < min_bal) {
  SOP("balance lesser than minimum balance, penalty imposed");
  balance -= penalty;
}
else {
  SOP("balance higher than minimum balance");
}
}
}
```

```
class cur_acct extends account {
 double balance;
 cur_acct(String n, int ac, String actype, double bl) {
 name = n;
 account_num = ac;
 actype = acc_type;
 balance = bl;
 }
 void deposit(int val) {
  balance += val;
 }

 void display_bal() {
   SOP("balance is: " + balance);
 }

 void deposit_interest() {
   SOP("current account doesnt provide any interest");
 }
 void withdraw(int val) {
   SOP("current account doesnt provide withdrawal facility");
 }
 void check_min() {
  SOP("no minimum balance requirement for current acc");
 }
}
```

```
class bank {
 PSVM(String args[]) {
 Scanner sc = new Scanner(System.in);
 SOP("enter your name, acc numb, acc type, balance");
String name = sc.next();
int account_num = sc.nextInt();
String acc_type = sc.next();
double balance = sc.nextDouble();
if (acc_type.equals("savings")) {
 sav_acct a1 = new sav_acct(name, accnt_num, acc_type, balance);
 int choice = 0;
while (chice != 6) {
SOP("1. deposit \n 2. display balance \n 3. compute and deposit
      interest \n 4. withdraw \n 5. check for minimum balance \n
        6. exit");
choice = sc.nextInt();
switch (choice) {
 case(1):
 SOP("enter the value to deposit");
 int val = sc.nextInt();
 a1.deposit(val);
 break;
 case(2):
 a1.display_bal();
 break;
 case(3):
 a1.deposit_interest;
 break;
```

```
case (4):
    SOP ("enter the value to withdraw");
    int withd = sc.nextint();
    a1.withdraw(withd);
    break;
case (5):
    a1.check_min();
    break;
case(6):
    SOP(" exited ");
    break;
default:
    SOP("enter valid choice");
    break;
  }
}
}
else {
curr-acct a1 = new cur-acct (name, account_num, acc_type, balance);
int choice = 0;
while (choice != 6) {
SOP("1. deposit \n 2. display \n 3. Compute & deposit interest \n 4. withdraw \n 5. check
      for min balance \n 6. exit");
choice = sc.nextInt();
switch (choice) {
case (1): SOP("enter value to deposit:");
      int val = sc.nextInt();
      a1.deposit(val);
      break;
```

```
case (2):
  a1. display_bal ();
  break;
case (3):
  a1. deposit_interest ();
  break;
case (4):
    SOP(" enter value to withdraw ");
    int withd = sc. nextint ();
    a1. withdraw (withd);
    break;
case (5):
    a1. check_min();
    break;
case (6):
    SOP(" exited ");
    break;
default:
    SOP (" enter valid choice ");
    break;
  }
 }
 }
 }
 }
```



O/P for savings

enter your name, account number, account type, balance
Zayd
10001
savings
100000
enter your choice (1. deposit 2. display balance 3. compute & deposit interest 4. withdraw 5. check for min bal 6. exit)
1
enter value to deposit : 1000
enter choice
2
Balance is 11000.
enter choice
3
enter time period
1
enter choice
2
Balance is 11550
enter choice
4
enter value to withdraw
1550
withdrawal successful
new balance : 10000
enter choice 5
balance higher than min balance
enter choice 6
exited

SSD
29/12/2022

O/P for current

enter your name, acc number, accout type, balance
Boyd
10002
Current
10000
enter your choice (1. deposit 2. display 3. compute & deposit
4. withdraw using cheque 5. check for min bal 6. exit)
1
enter value to deposit: 2000
enter choice
2
Balance is: 12000
enter choice
3
current acc doesnt provide any interests
enter choice
4
enter value to withdraw: 150000
withdrawal successful
new balance: -3000
enter choice
5
balance lesser than min bal, penalty imposed
enter choice
2
new balance: -3100
enter choice
6
exited

SS
29/12/20

—x—

Output for savings:

```
enter your name, account number, aacount type(savings/current), balance
zayd
10000001
savings
10000
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
1
enter the value to deposit
1000
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 11000.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
3
enter the time period
2
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 12127.5
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
4
enter the value to withdraw
1000
withdrawal successful
new balance: 11127.5
1.deposit
```

```
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
4
enter the value to withdraw
1000
withdrawal successful
new balance: 11127.5
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 11127.5
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
5
balance higher than minimum balance
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
6
exited
```

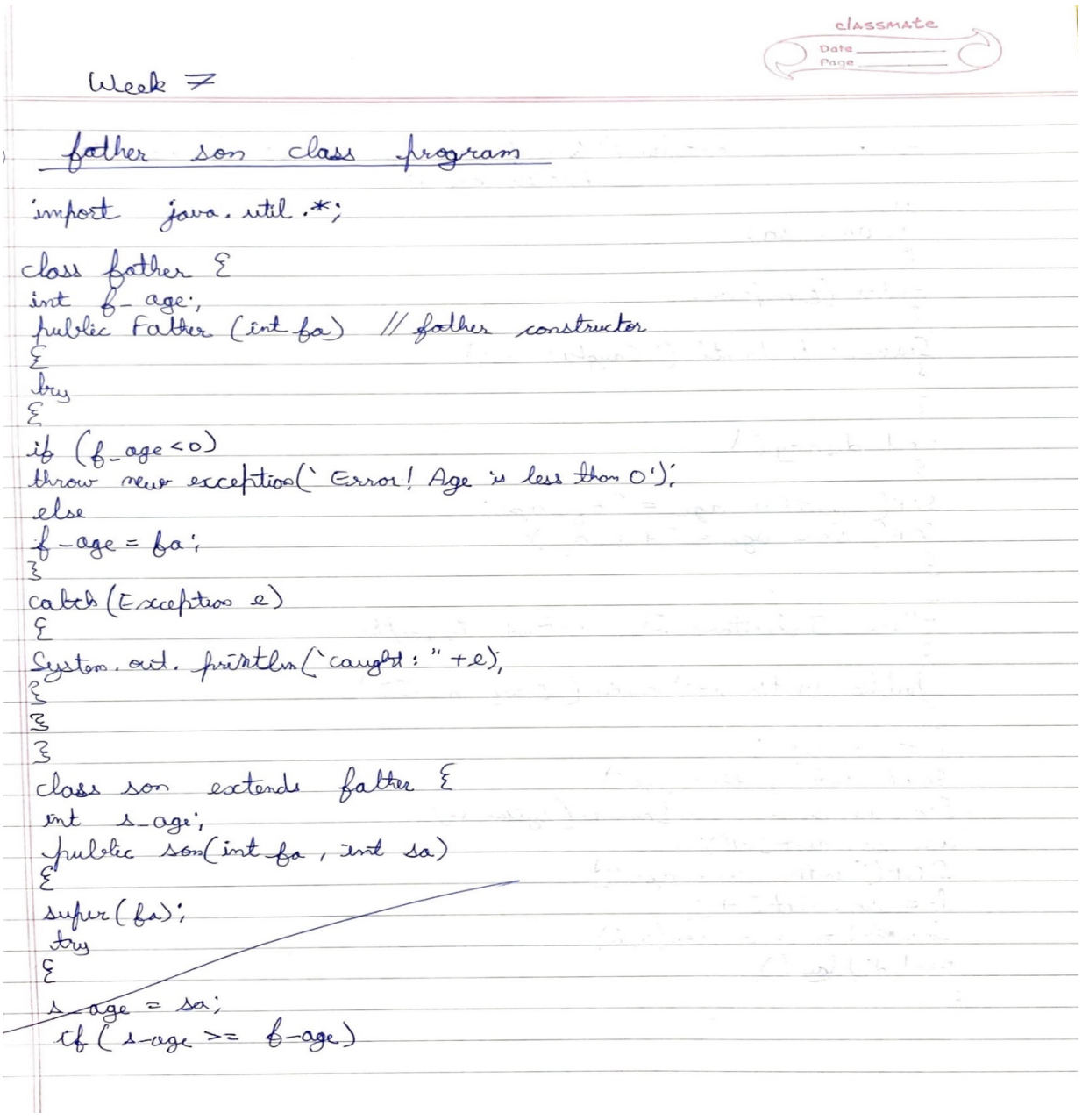
Output for current:

```
enter your name, account number, aacount type(savings/current), balance
zayd
10000000
current
10000
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
1
enter the value to deposit
10
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
2
Balance is: 10010.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
3
Current account doesnt provide any interest
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
4
enter the value to withdraw
200000
withdrawal successful
new balance: -189990.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
5
```

```
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
5
balance lesser than minimum balance, penalty imposed
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
2
Balance is: -190090.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
6
exited
PS C:\Users\Admin\Desktop\zayd>
```

LAB PROGRAM 7:

Q: Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father’s age.



Week 7

father son class program

```
import java.util.*;
class father {
int f_age;
public Father (int fa) // father constructor
{
try
{
if (f_age<0)
throw new exception(" Error! Age is less than 0");
else
f_age = fa;
}
catch (Exception e)
{
System.out.println("caught: " +e);
}
}
}
class son extends father {
int s_age;
public son(int fa, int sa)
{
super(fa);
try
{
s_age = sa;
if (s_age >= f_age)
```

```
throw new Exception("Error! sons age cannot be more than
                                fathers age");
else
s_age = sa;
}
catch (Exception e)
{
System.out.println("Caught: "+e);
}
}
void display()
{
SOP("fathers age = "+f_age);
SOP("sons age = "+s_age);
}
}
class InheritanceTree extends Exception
{
public static void main(String args[])
{
int a,b;
SOP('enter fathers age');
Scanner sc = new Scanner(System.in);
a = sc.nextInt();
SOP("enter sons age");
b = sc.nextInt();
Son ob1 = new son(a,b);
ob1.display();
```

O/P

enter fathers age 20
enter sons age 30
caught: java.lang.Exception: Error! sons age cannot be more than fathers age.

enter fathers age 30
enter sons age 20
Fathers age: 30
Sons age: 20

enter fathers age 0
enter sons age -1
caught: java.lang.Exception: Error! sons age is less than 0

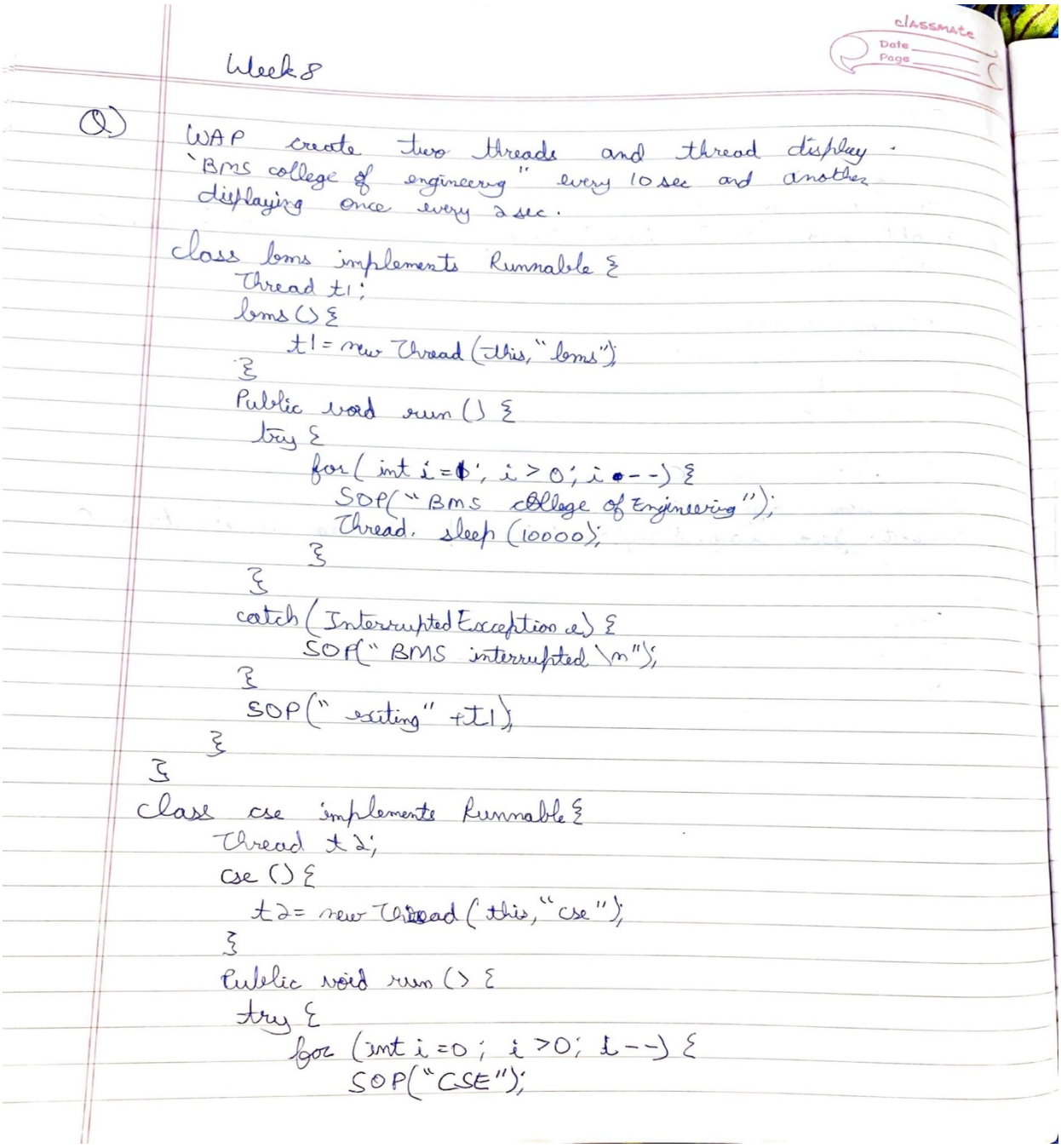
5/10/2023

Output:

```
C:\Users\Admin\Desktop\f7zz>javac InheritanceTree.java

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
25
Enter the son's age
23
Father's age = 25
Son's age = 23

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
25
Enter the son's age
26
Caught : java.lang.Exception: Error! Son's age cannot be more than the Father's age
Father's age = 25
Son's age = 26

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
0
Enter the son's age
0
Caught : java.lang.Exception: Error! Son's age cannot be more than the Father's age
Father's age = 0
Son's age = 0

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
0
Enter the son's age
-1
Caught : java.lang.Exception: Error! Son's age is less than 0
Father's age = 0
Son's age = -1

C:\Users\Admin\Desktop\f7zz>
```

LAB PROGRAM 8:

Q: Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.



Week 8

Q) WAP create two threads and thread display "BMS college of engineering" every 10 sec and another displaying once every 2 sec.

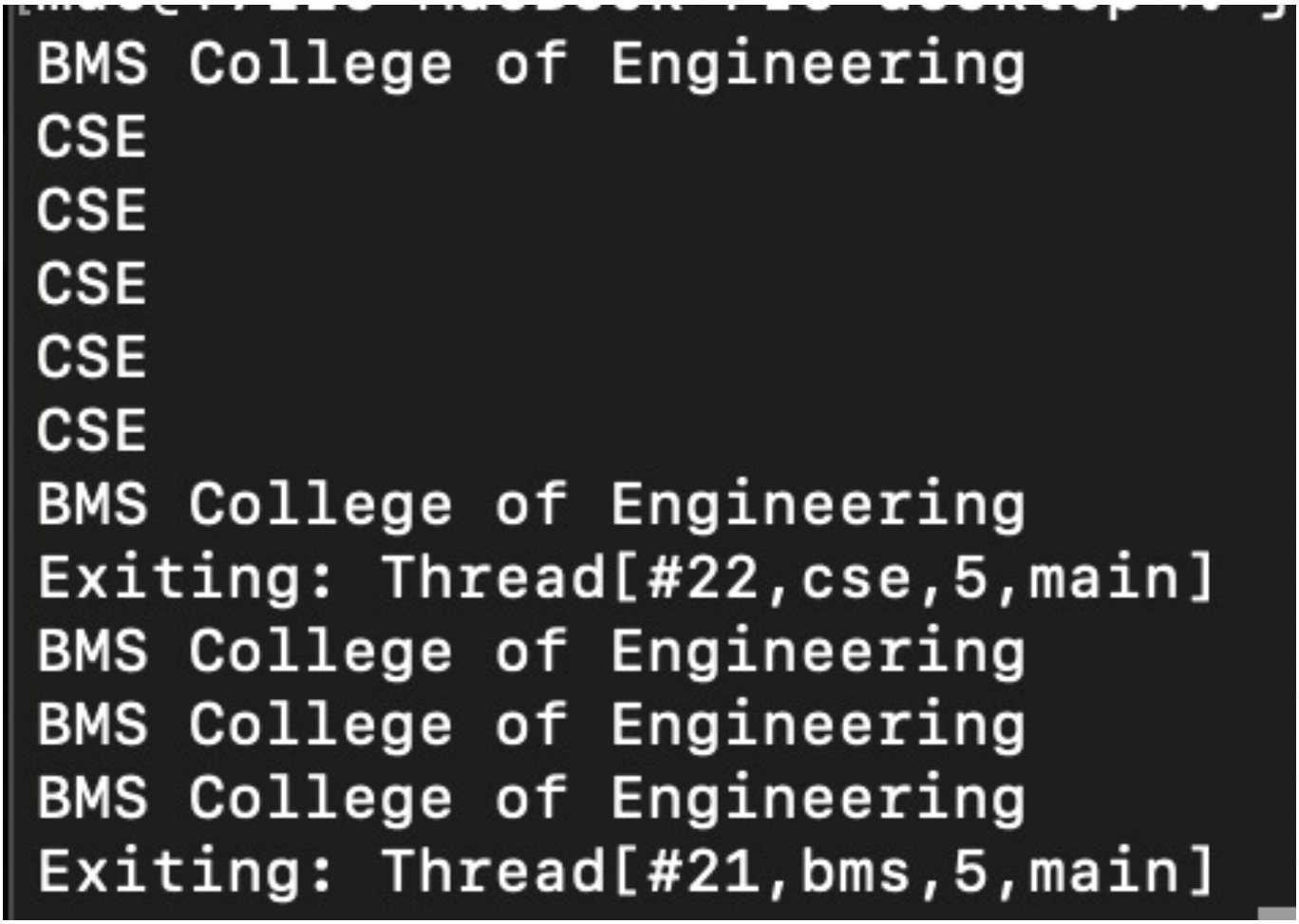
```
class bms implements Runnable {
    Thread t1;
    bms() {
        t1 = new Thread(this, "bms");
    }
    Public void run() {
     try {
         for( int i = 0; i > 0; i--) {
            SOP("BMS College of Engineering");
            Thread.sleep(10000);
         }
     }
     catch(InterruptedException e) {
         SOP("BMS interrupted \n");
     }
     SOP("exiting" +t1);
    }
}
class cse implements Runnable {
    Thread t2;
    cse() {
      t2 = new Thread(this, "cse");
    }
    Public void run() {
     try {
        for (int i = 0; i > 0; i--) {
            SOP("CSE");
```

```
            Thread.sleep(2000);
        }
    }
    catch (InterruptedException e) {
            SOP("CSE interrupted\n");
    }
    SOP("Exiting : " +t2);
  }
}
class threadprg {
    PSVM(String args[]) {
      bms obj1 = new bms();
      cse obj2 = new cse();
      obj1.t1.start();
      obj2.t2.start();
    }
}
```

O/P

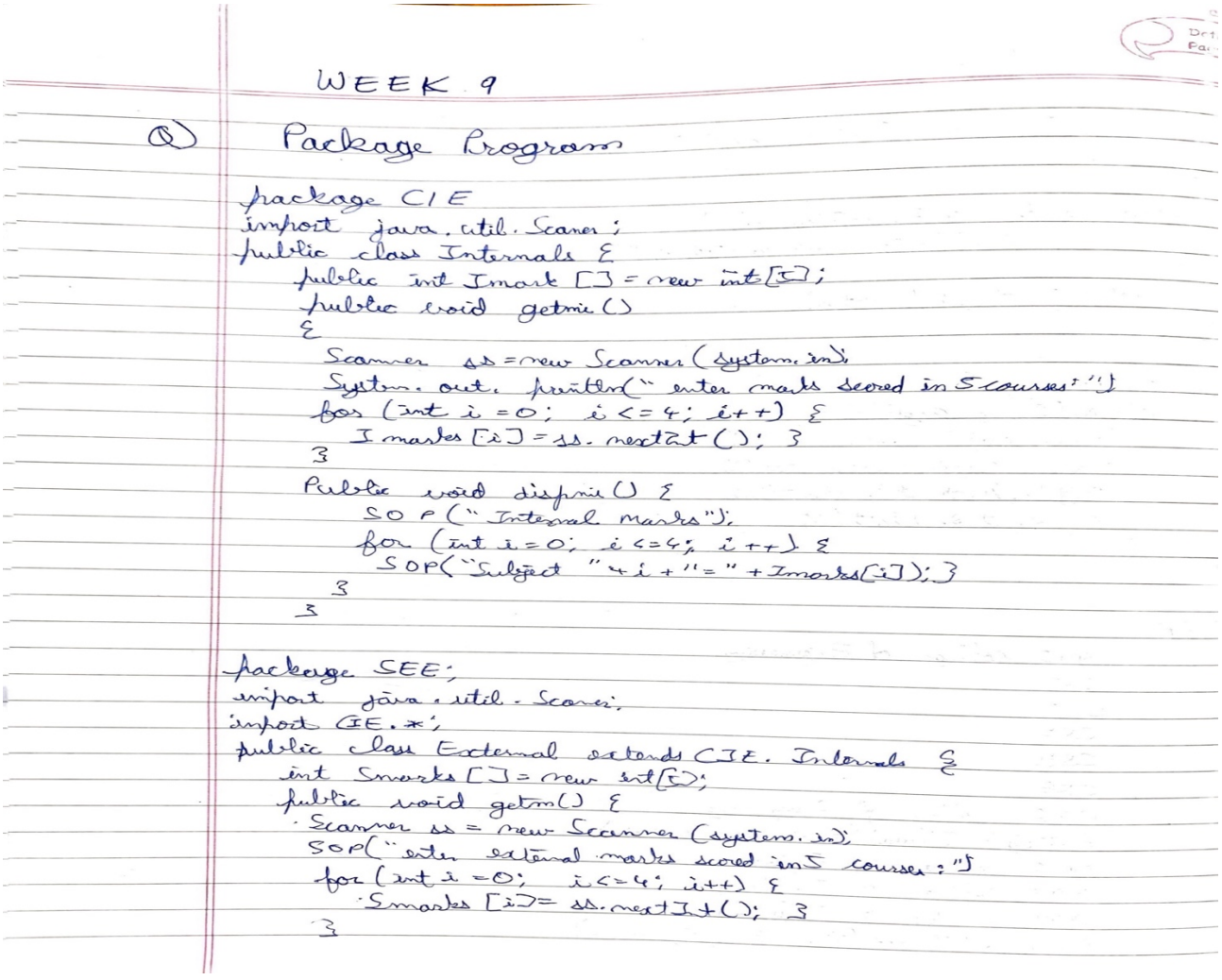
```
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
Exiting : Thread(#22, cse, 5, main)
Bms college of Engineering
BMS college of Engineering
BMS college of Engineering
Exiting : Thread(#21, bms, 5, main)
```

Output:


BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
Exiting: Thread[#22,cse,5,main]
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
Exiting: Thread[#21,bms,5,main]

LAB PROGRAM 9:

Q: Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.



WEEK 9

Q) Package Program

```
package CIE
import java.util.Scaner;
public class Internals {
    public int Imark [] = new int[5];
    public void getmie()
    {
      Scanner ss=new Scanner(System.in);
      System.out.println(" enter marks scored in 5 courses:");
      for (int i =0; i<=4; i++) {
        I marks [i] = ss.nextInt(); }
      }
      Public void displie() {
          SOP(" Internal Marks");
          for (int i=0; i<=4; i++) {
            SOP("Subject "+i+"=" +Imarks[i]); }
        }
    }

package SEE;
import java.util.Scaner;
import CIE.*;
public class External extends CIE.Internals {
    int Smarks [] = new int[5];
    public void getsm() {
      Scanner ss = new Scanner(System.in);
      SOP(" enter external marks scored in 5 courses:")
      for (int i =0; i<=4; i++) {
        Smarks [i]= ss.nextInt(); }
      }
```

```
public void displm() {
    SOP("external marks");
    for(int i=0; i<=4; i++) {
      SOP("subject" +i + "=" + Smarks[i]); }
    }
    public void finalcal() {
    int Final[] = new int[5];
      for(int j=0; j<=4; j++) {
        Final[j] = Imarks[j] + (Smarks[j]/2); }
        SOP("Final marks");
    for(int i=0; i<=4; i++) {
      SOP("subject" + i + "=" + Final[i]); }
    }
}
```

```
package CIE
import java.util.Scanner;
public class Student {
    public String usn, name;
    public int sem;
    public void getd() {
      Scanner s = new Scanner(System.in);
      SOP("enter USN, NAme & sem");
      usn = s.next();
      name = s.next();
      sem = s.nextInt();
    }
    public void disp() {
    SOP("\n Student Details:\n USN:" +usn + "\n Name:" + name + "\n SEM:" + sem);
    }
}
```

```
import CIE.*;
import SEE.*;

class P_main {
    PSVM(String args[]) {
     Student s1 = new Student();
     s1.getd();
     s1.disp();
     External e1 = new External();
     e1.getmai();
     e1.dispmi();
     e1.getmi();
     e1.dispm();
     e1.finalcal();
    }
}
```

O/P

```
Enter USN, Name & Sem
1BM21CS254
Name: Zayd
 Sem: 3
Enter marks scores in 5 scores:
45
48
42
41
40

Internals Marks
 Subject 0 = 43
```

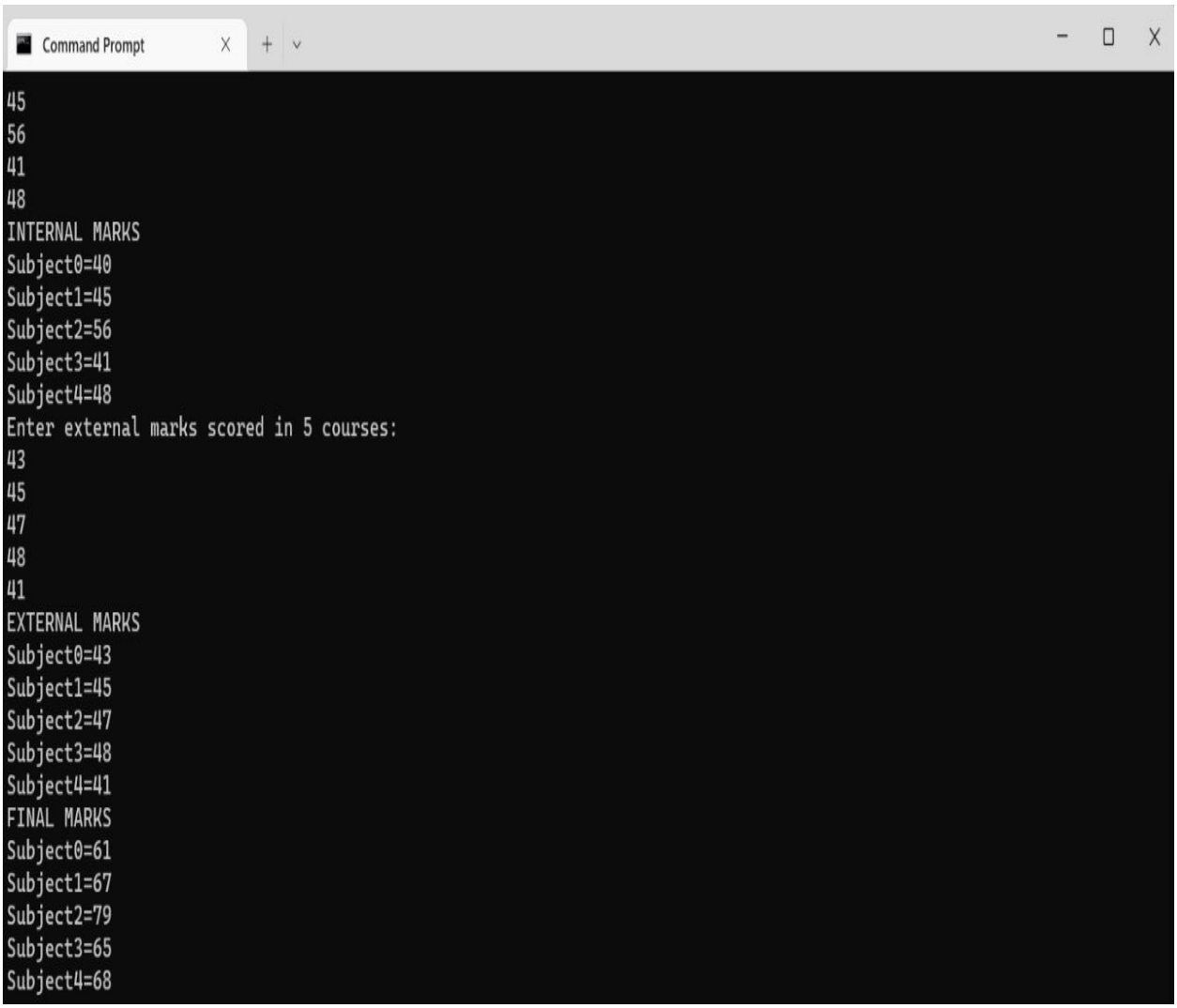
Subject 1 = 48
Subject 2 = 42
Subject 3 = 41
Subject 4 = 40
Enternal Marks score for 5 course
49
48
47
46
45

External Mark
"
"

Final marks
Subject 0 : 69
Subject 1 : 72
Subject 2 : 65
Subject 3 : 64
Subject 4 : 62

Output:

```
Enter USN, NAME & SEM
1BM21CS254
Z
1

Student Details:
USN:1BM21CS254
NAME:Z
SEM:1
Enter marks scored in 5 courses:
40
45
56
41
48
INTERNAL MARKS
Subject0=40
Subject1=45
Subject2=56
Subject3=41
Subject4=48
Enter external marks scored in 5 courses:
```


Command Prompt
45
56
41
48
INTERNAL MARKS
Subject0=40
Subject1=45
Subject2=56
Subject3=41
Subject4=48
Enter external marks scored in 5 courses:
43
45
47
48
41
EXTERNAL MARKS
Subject0=43
Subject1=45
Subject2=47
Subject3=48
Subject4=41
FINAL MARKS
Subject0=61
Subject1=67
Subject2=79
Subject3=65
Subject4=68